EMO DETECT: Real time Emotion Detection from Faces, Text and Live Webcam

MINIPROJECT REPORT

Submittedby

THIRUVEEDULA SUJITH KUMAR
[RA2011047010145]

Undertheguidanceof
Dr. SUMATHY G
(GuideAffiliation)
Assistant Professor

Department of Computational Intelligence

FACULTY OF ENGINEERING AND TECHNOLOGY

SCHOOL OF COMPUTING

SRM INSTITUTE OF SCIENCE AND TECHNOLOGY

Kattankulathur, Kancheepuram

MAY 2023

SRMINSTITUTEOF SCIENCEANDTECHNOLOGY

(UnderSection3ofUGCAct,1956)

BONAFIDECERTIFIC ATE

Certifiedthat18AIE324T_Cognitive Science and Analytics titled "**EMOTION ANALYSIS WITH STREAMLIT**" is the bonafide work of "**THIRUVEEDULA SUJITH KUMAR [RA2011047010145]**"who carried out theminor projectwork under my supervision. Certified further, that to the best of myknowledge the work reported herein does not form any other project report ordissertation on the basis of which a degree or award was conferred on an earlieroccasiononthisoranyother candidate.

SIGNATURE

Dr.SUMATHY G
GUIDE
Assistant Professor

SIGNATURE

Dr. Annie Uthra
HEADOFTHEDEPARTMENT
Professor
Dept. OfComputationalIntelligence

ABSTRACT

In this project, we propose to develop an emotion detection system using machine learning algorithms and natural language processing techniques. Our system will leverage existing datasets of sentiment-bearing words and phrases, as well as develop its own models through supervised learning on annotated data. The goal of our system is to accurately identify the emotional tone of a piece of text, whether it is positive, negative, or neutral, and provide valuable insights into people's opinions and attitudes towards a particular topic or product.

INDEX

S.No	PROJECT CONTENT	PAGE NO.
A	Abstract	
B	Table Of Contents	
C	List Of Figures	
D	List of tables	
1	Introduction	
2	Literature Survey	
3	Proposed Work	
4	Design Methodology	
5	Usecase Diagram	
6	Class Diagrams With Description	
7	Sequence Diagrams	
8	Collaboration Diagrams	
9	Package Diagram	
10	Component Diagram	
11	Deployment Diagram	
12	Module And Algorithm Description	
13	Coding	
14	Result And Conclusion	
15	Reference	
16	Appendix Output Screenshot	

LISTOFFIGURES

FigureNo. **FigureName** **PageNo.**

3.1 System Architecture 13
3.2 Accuracy Comparision 16
3.3 Confusion Matrix 17

LISTOFTABLES

TableNo.	TableName	PageNo.
2.1	Literature Survey	11-12

ABBREVIATIONS

LSTM	Long short-term memory
CNN	ConvolutionalNeuralNetwork
UI	UserInterface
API	Application Programming Interface

CHAPTER1

INTRODUCTION

Sentiment analysis in text, also known as opinion mining, is a natural language processing technique that involves the use of machine learning algorithms and statistical models to identify and extract subjective information from text. The goal of sentiment analysis is to determine the emotional tone or attitude expressed in a piece of text, whether it is positive, negative, or neutral.

CHAPTER 2

LITERATURE SURVEY

Over the past few years, study the human sentiment has been researched upon through various automation techniques. Following the development of Natural Language Processing, textual analysis has been made significantly more accessible with the help of Neural Networks and their types. Studies suggest, that LSTMs and CNNs are the two most popular choices for performing sentiment analysis on behavioral data collected from social media. [3] explores the use of Convolutional Neural Networks for the purpose of Twitter Sentiment Analysis upon the SemiEval Dataset with an aim of feature extraction from user behavior information and handling a huge amount of unstructured data. In [1] we found recommending the places which is near to the user's current location by analyzing the different reviews and consequently computing the score grounded on it. In [4] logistic regression has the highest accuracy among the other machine learning algorithms they used in research for predicting frustration.

Title	**Author**	**Year**	**Findings**	**Limitations**
Sentiment Analysis Based on Deep Learning Approaches [IEEE]	Kaur, J., & Sidhu, B. K.	2018	Gives an overview on sentiment analysis and what different methods and levels which are used with it	Low Accuracy in models
Amrita-CEN-SentiDB1:Improved Twitter Dataset for Sentimental Analysis and Application of Deep learning [IEEE]	Naveenkumar K S; Vinayakumar R; Soman K P	2019	Evaluates the performance of linear and nonlinear text representation methods for sentimental analysis	Output is dependent on decision making Vectors
Deep Learning for Aspect-Based Sentiment Analysis: A Comparative Review [ACM]	Do, H.H.; Prasad, P.; Maag, A.; Alsadoon, A.J.	2019	Aspect extraction and sentiment classification	Deep Learning Models are computationally expensive.

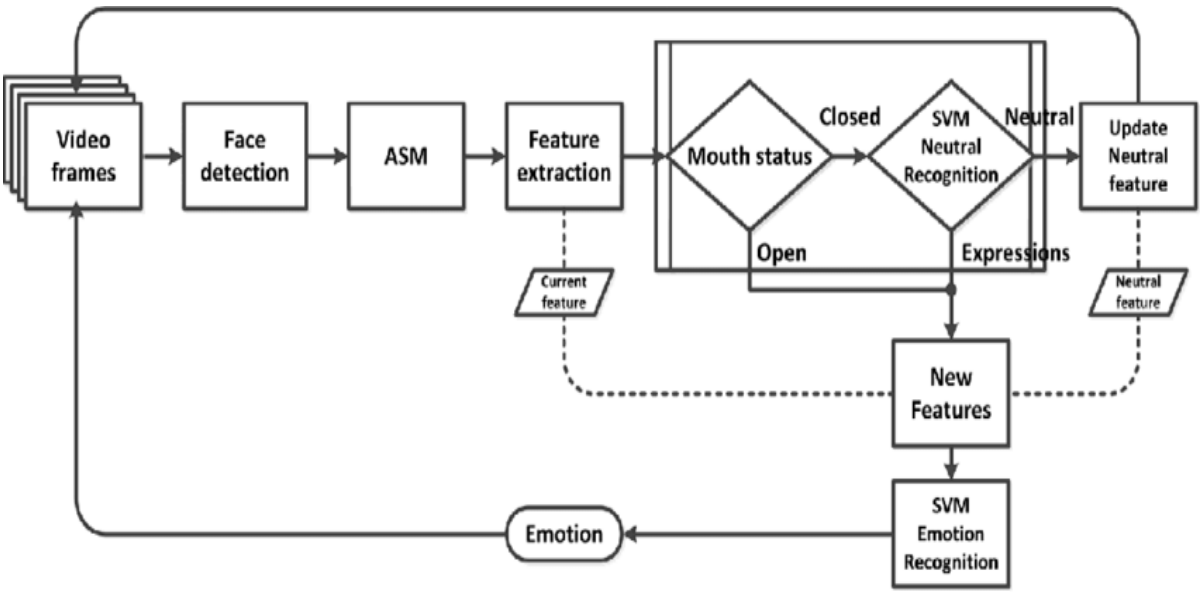
Sentiment Analysis of Uber & Ola using Deep Learning. 2020 (ICOSEC). [IEEE]	Indulkar, Y., & Patil, A.	2020	The CNN model was moderate concerning the DNN, which was questioning	The accuracy for the Ola tweets was not that good as compared to the Uber though cleaning was done still it couldn't generate the accuracy as expected.
Sentimental analysis using fuzzy and naive bayes. [IEEE]	Mehra, R., Bedi, M. K., Singh, G., Arora,	2018	Applied a hybrid of naive Bayes and Fuzzy classifier to this set conduct sentiment analysis	Work only on twitter size which is 140 words.
Semi-supervised dimensional sentiment analysis with variational autoencoder [ELSEVIER]	Wu, C.; Wu, F.; Wu, S.; Yuan, Z.; Liu, J.; Huang, Y	2019	Encoding, sentiment prediction, and decoding	Computationally expensive
Application of deep learning to sentiment analysis for recommender system on cloud [IEEE]	Preethi, G.; Krishna, P.V.; Obaidat, M.S.; Saritha, V.; Yenduri, S	2018	Recommending the places that are near to the user's current location by analyzing the different reviews and consequently computing the score grounded on it	RDSA is not done with variety of datasets and also with more volume of data.
Twitter sentiment analysis using deep learning methods [IEEE]	Ramadhani, A.M.; Goo, H.S.	2018	Handling a huge amount of unstructured data	Computationally Expensive for huge data.
Frustration Detection On Reviews Using Machine Learning [IEEE]	Suri, S., Sharma, K., &Papneja, S	2020	We conclude that Logistic Regression has the highest accuracy among the different machine learning algorithms we used for predicting frustration.	Only considering the negative reviews for frustration detection.
Sentiment analysis using deep learning on Persian texts [IEEE]	Roshanfekr, B.; Khadivi, S.; Rahmati, M.	2018	Evaluating deep learning methods using the Persian language	The use of word vector representations which is done in an unsupervised way.

CHAPTER 3

PROPOSED WORK

We have tried to increase the accuracy as much as possible as many of the transfer learning algorithm which involve pretrained models works well but not that much accurate. So, our main objective is to develop a deep learning model that can detect the emotions of the content with more accuracy and performance.The main idea behind this project is to integrate artificial intelligence with popular Streamlit software. In this project we will do sentimental analysis on the emails which we will get from our outlook account using robotic process automation tool i.e., Streamlit and return the percentage of emotions on the web portal.

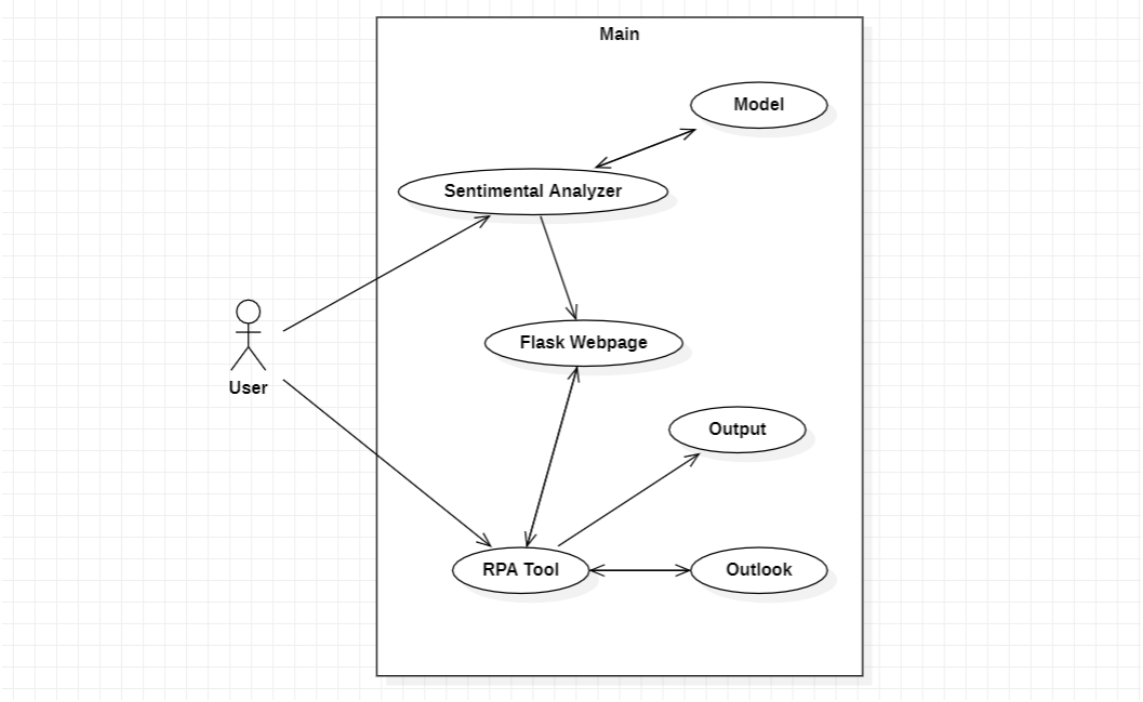
Fig 3.1



CHAPTER 4

MODULE DESCRIPTION

1. Data Collection
2. Pre-processing Module
3. Feature Extraction Module
4. Classification Module
5. Integration Module



Real time Emotion Detection using text, face and live webcam.

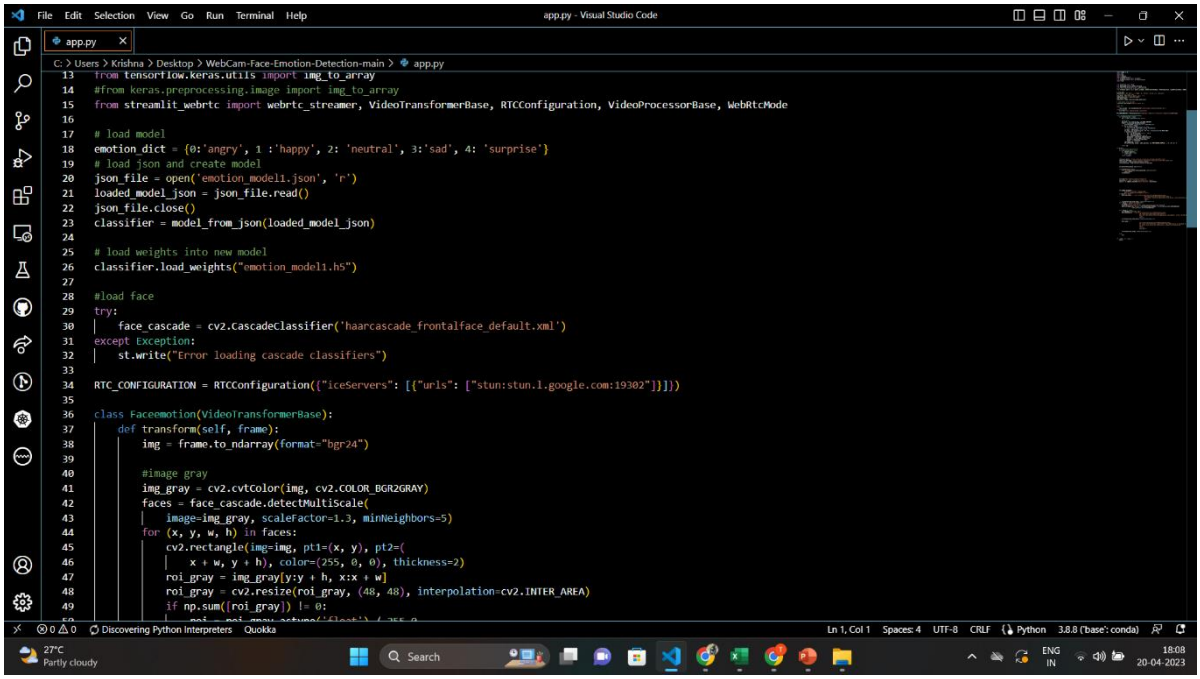
VADER Sentiment Analysis. VADER (Valence Aware Dictionary and sEntiment Reasoner) is a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media, and works well on texts from other domains.

A powerful NLP library. Flair allows you to apply our state-of-the-art natural language processing (NLP) models to your text, such as named entity recognition (NER), sentiment analysis, part-of-speech tagging (PoS), special support for biomedical data, sense disambiguation and classification, with support for a rapidly growing number of languages.

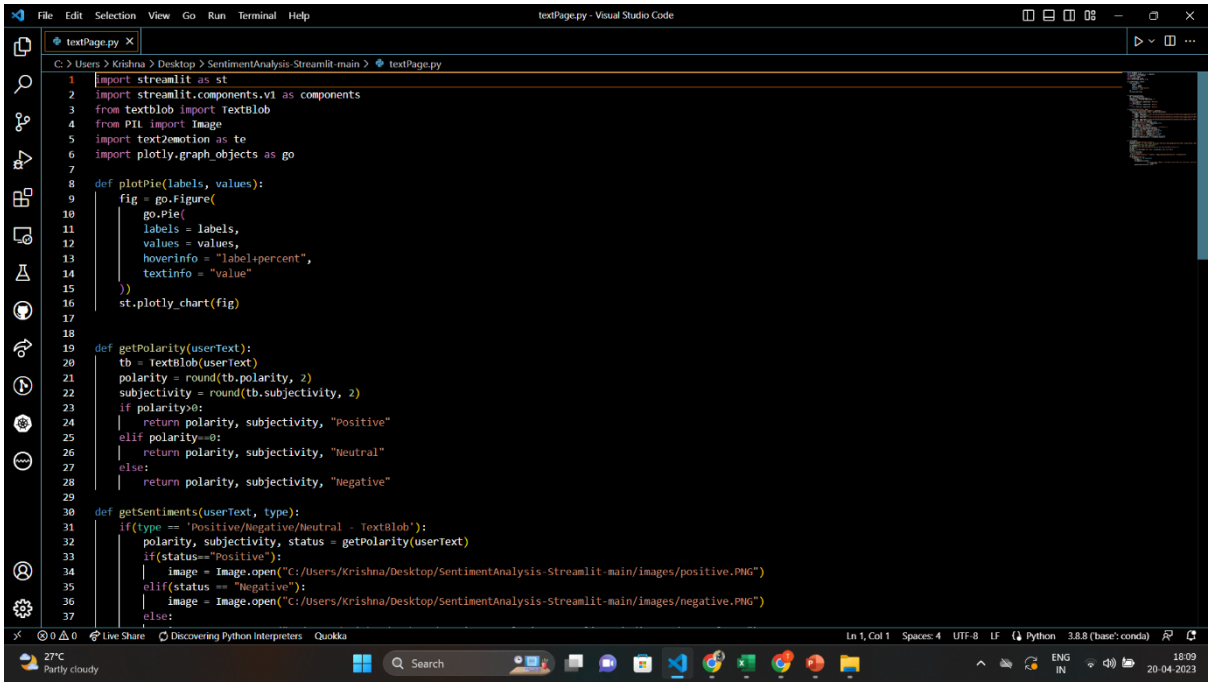
A text embedding library. Flair has simple interfaces that allow you to use and combine different word and document embeddings, including our proposed Flair embeddings and various transformers.

CHAPTER 5

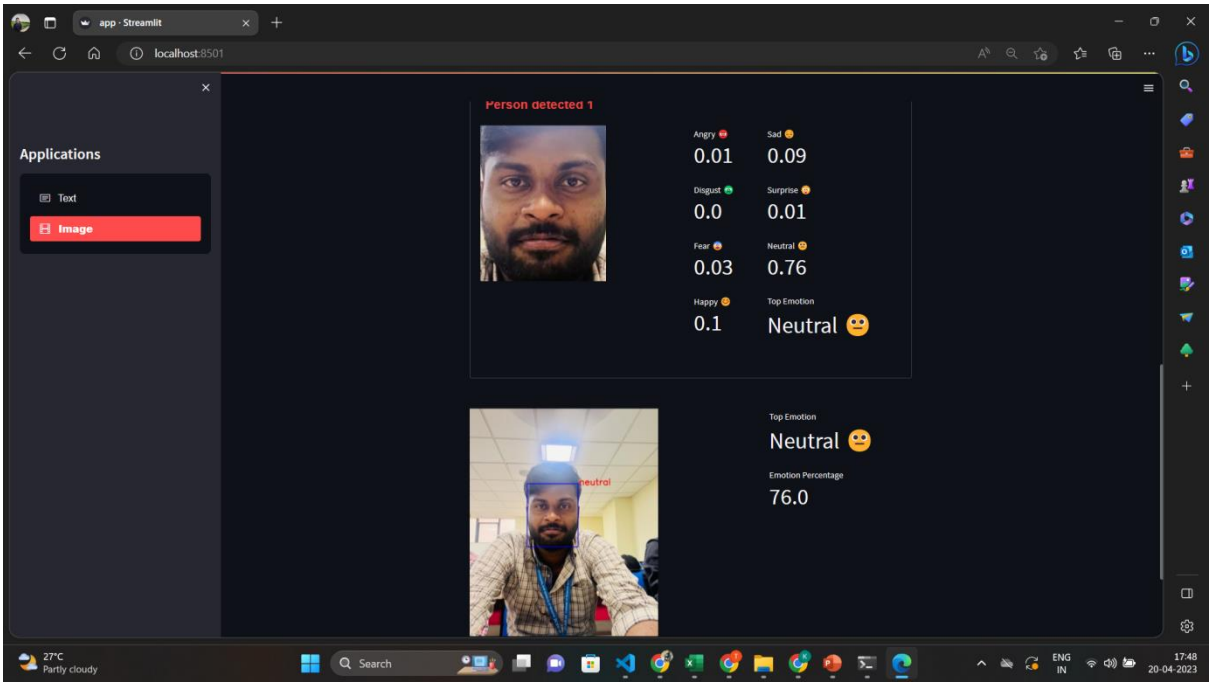
RESULT AND CONCLUSION

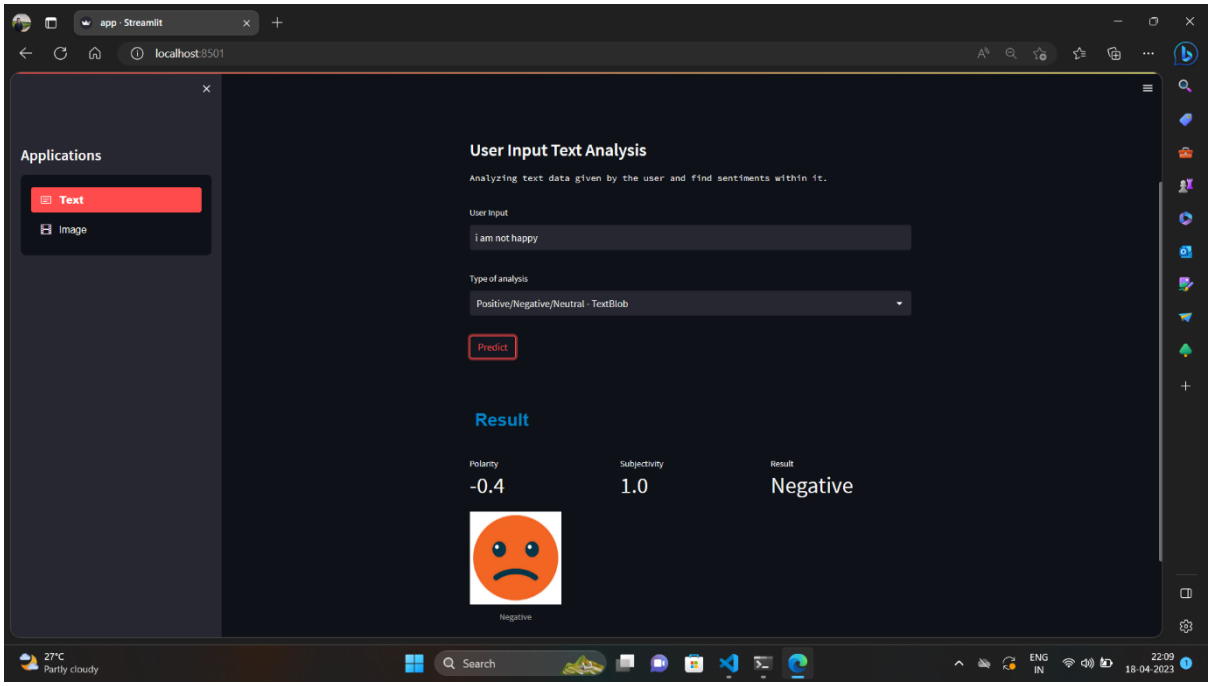


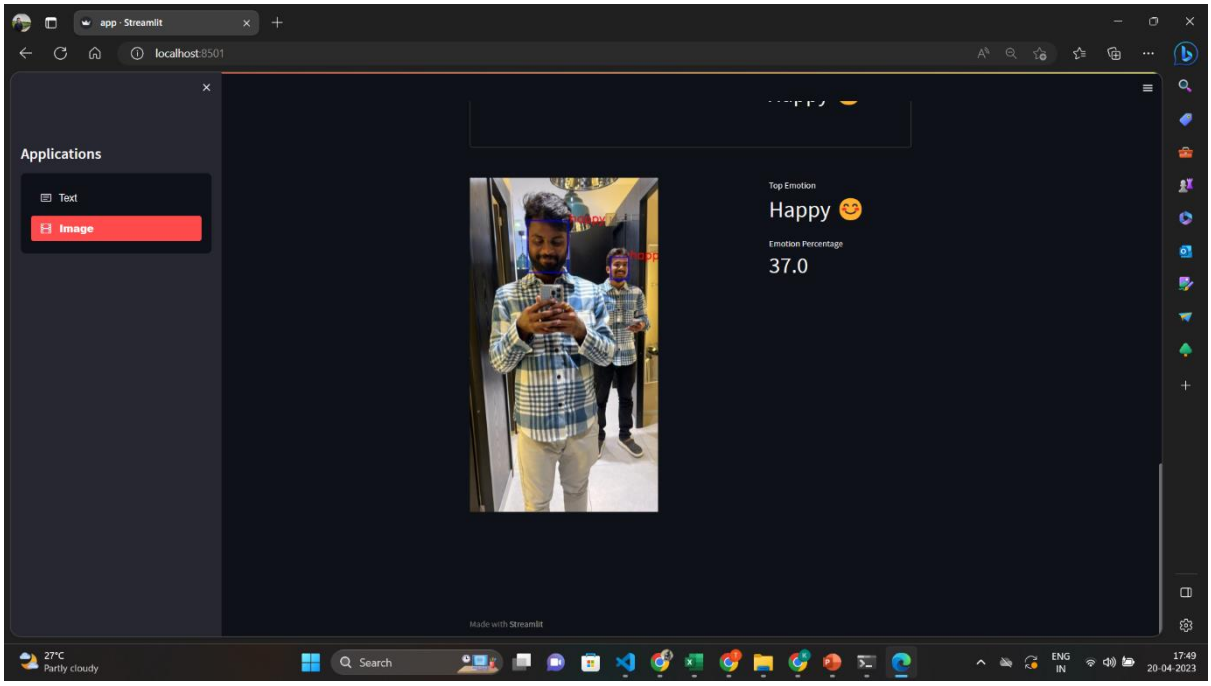
```python
from tensorflow.keras.utils import img_to_array
#from keras.preprocessing.image import img_to_array
from streamlit_webrtc import webrtc_streamer, VideoTransformerBase, RTCConfiguration, VideoProcessorBase, WebRtcMode

# load model
emotion_dict = {0:'angry', 1 :'happy', 2: 'neutral', 3:'sad', 4: 'surprise'}
# load json and create model
json_file = open('emotion_model1.json', 'r')
loaded_model_json = json_file.read()
json_file.close()
classifier = model_from_json(loaded_model_json)

# load weights into new model
classifier.load_weights("emotion_model1.h5")

#load face
try:
    face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
except Exception:
    st.write("Error loading cascade classifiers")

RTC_CONFIGURATION = RTCConfiguration({"iceServers": [{"urls": ["stun:stun.l.google.com:19302"]}]})

class Faceemotion(VideoTransformerBase):
    def transform(self, frame):
        img = frame.to_ndarray(format="bgr24")

        #image gray
        img_gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
        faces = face_cascade.detectMultiScale(
            image=img_gray, scaleFactor=1.3, minNeighbors=5)
        for (x, y, w, h) in faces:
            cv2.rectangle(img=img, pt1=(x, y), pt2=(
                x + w, y + h), color=(255, 0, 0), thickness=2)
            roi_gray = img_gray[y:y + h, x:x + w]
            roi_gray = cv2.resize(roi_gray, (48, 48), interpolation=cv2.INTER_AREA)
            if np.sum([roi_gray]) != 0:
```

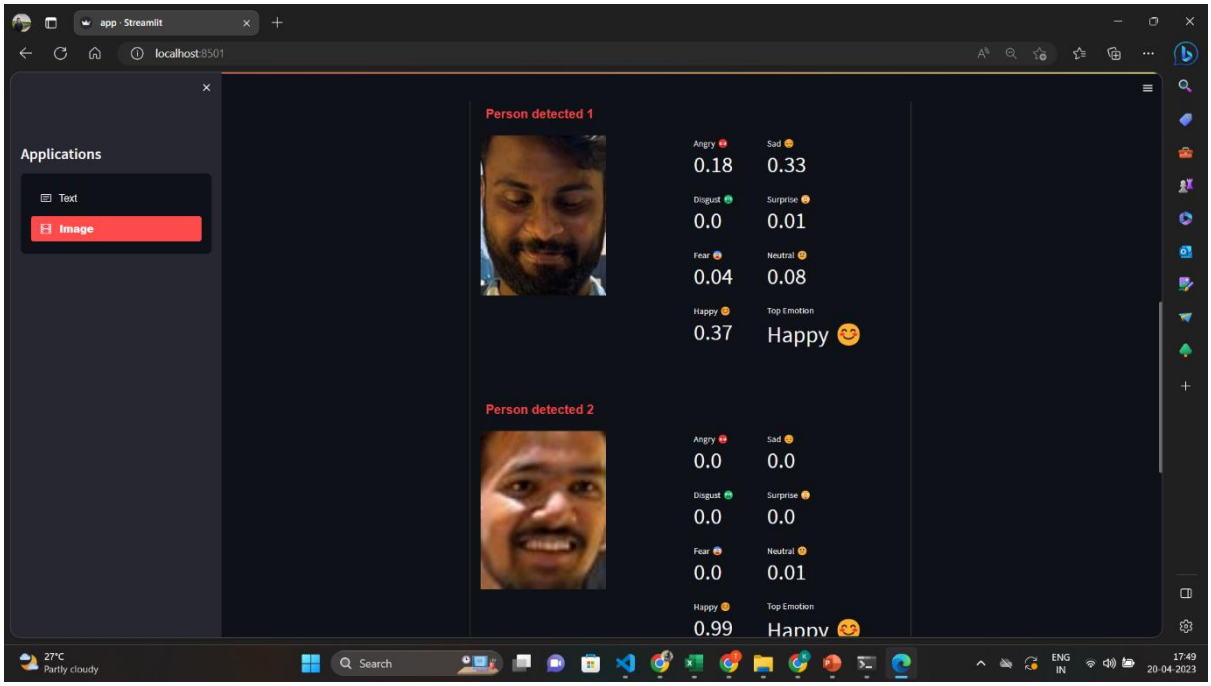


```python
import streamlit as st
import streamlit.components.v1 as components
from textblob import TextBlob
from PIL import Image
import text2emotion as te
import plotly.graph_objects as go

def plotPie(labels, values):
    fig = go.Figure(
        go.Pie(
        labels = labels,
        values = values,
        hoverinfo = "label+percent",
        textinfo = "value"
    ))
    st.plotly_chart(fig)


def getPolarity(userText):
    tb = TextBlob(userText)
    polarity = round(tb.polarity, 2)
    subjectivity = round(tb.subjectivity, 2)
    if polarity>0:
        return polarity, subjectivity, "Positive"
    elif polarity==0:
        return polarity, subjectivity, "Neutral"
    else:
        return polarity, subjectivity, "Negative"

def getSentiments(userText, type):
    if(type == 'Positive/Negative/Neutral - TextBlob'):
        polarity, subjectivity, status = getPolarity(userText)
        if(status=="Positive"):
            image = Image.open("C:/Users/Krishna/Desktop/SentimentAnalysis-Streamlit-main/images/positive.PNG")
        elif(status == "Negative"):
            image = Image.open("C:/Users/Krishna/Desktop/SentimentAnalysis-Streamlit-main/images/negative.PNG")
        else:
```


app · Streamlit
localhost:8501
Applications
Text
Image
Person detected 1
Angry
0.01
Sad
0.09
Disgust
0.0
Surprise
0.01
Fear
0.03
Neutral
0.76
Happy
0.1
Top Emotion
Neutral
Top Emotion
Neutral
Emotion Percentage
76.0
27°C
Partly cloudy
Search
ENG
IN
17:48
20-04-2023


app · Streamlit
localhost:8501
Applications
Text
Image
User Input Text Analysis
Analyzing text data given by the user and find sentiments within it.
User Input
i am not happy
Type of analysis
Positive/Negative/Neutral - TextBlob
Predict
Result
Polarity
-0.4
Subjectivity
1.0
Result
Negative
Negative
27°C
Partly cloudy
Search
ENG
IN
22:09
18-04-2023


app · Streamlit
localhost:8501
Applications
Text
Image
Top Emotion
Happy
Emotion Percentage
37.0
Made with Streamlit
27°C
Partly cloudy
Search
ENG
IN
17:49
20-04-2023


app · Streamlit
localhost:8501
Applications
Text
Image
Person detected 1
Angry
0.18
Sad
0.33
Disgust
0.0
Surprise
0.01
Fear
0.04
Neutral
0.08
Happy
0.37
Top Emotion
Happy
Person detected 2
Angry
0.0
Sad
0.0
Disgust
0.0
Surprise
0.0
Fear
0.0
Neutral
0.01
Happy
0.99
Top Emotion
Happy
27°C
Partly cloudy
Search
ENG
IN
17:49
20-04-2023

REFERENCES

1. Preethi, G.; Krishna, P.V.; Obaidat, M.S.; Saritha, V.; Yenduri, S. Application of deep learning to sentiment analysis for recommender system on cloud. In Proceedings of the 2017 International Conference on Computer, Information and Telecommunication Systems (CITS), Dalian, China, 21–23 July 2017; pp. 93–97. Electronics 2020, 9, 483 27 of 29

2. Suri, S., Sharma, K., &Papneja, S. (2020). *Frustration Detection On Reviews Using Machine Learning. 2020 International Conference for Emerging Technology (INCET).*

3. Naveenkumar K S; Vinayakumar R; Soman K P Amrita-CEN-SentiDB1:Improved Twitter Dataset for Sentimental Analysis and Application of Deep learning

4. Kaur, J., & Sidhu, B. K. (2018). Sentiment Analysis Based on Deep Learning Approaches. 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS).

5. Indulkar, Y., & Patil, A. Indulkar, Y., & Patil, A. (2020). Sentiment Analysis of Uber & Ola using Deep Learning. 2020 (ICOSEC).

6. Roshanfekr, B.; Khadivi, S.; Rahmati, M. Sentiment analysis using deep learning on Persian texts. In Proceedings of the 2017 Iranian Conference on Electrical Engineering (ICEE), Tehran, Iran, 2–4 May 2017; pp. 1503–1508.

7. Ramadhani, A.M.; Goo, H.S. Twitter sentiment analysis using deep learning methods. In Proceedings of the 2017 7th International Annual Engineering Seminar (InAES), Yogyakarta, Indonesia, 1–2 August 2017; pp. 1–4.

8. Yang, C.; Zhang, H.; Jiang, B.; Li, K.J. Aspect-based sentiment analysis

with alternating coattention networks. Inf. Process. Manag. 2019, 56, 463–478.

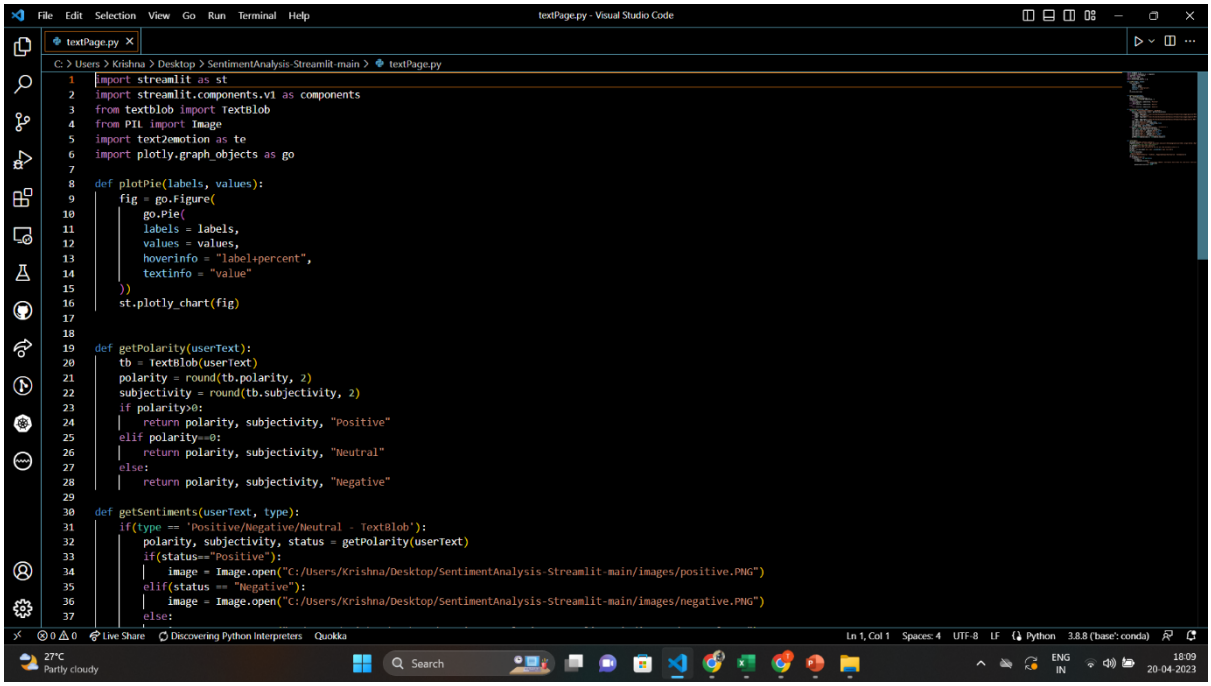
9. Do, H.H.; Prasad, P.; Maag, A.; Alsadoon, A.J. Deep Learning for Aspect-Based Sentiment Analysis: A Comparative Review. Expert Syst. Appl. 2019, 118, 272–299.

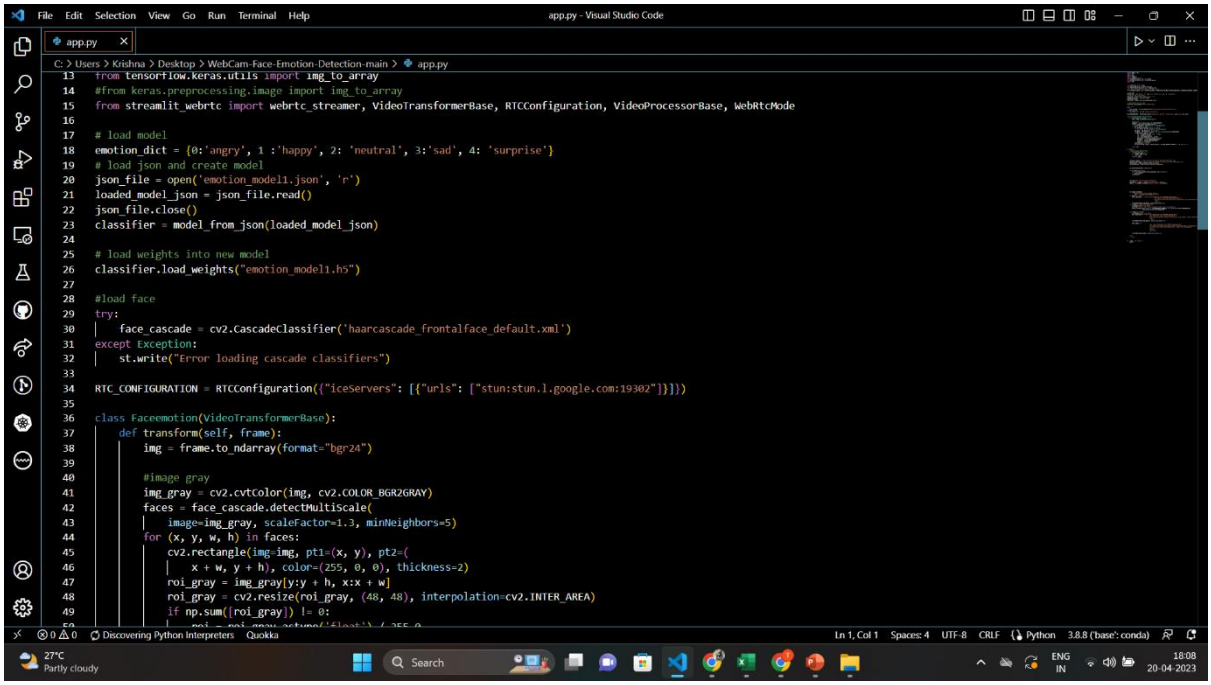
10. Wu, C.; Wu, F.; Wu, S.; Yuan, Z.; Liu, J.; Huang, Y. Semi-supervised dimensional sentiment analysis with variational autoencoder. Knowl. Based Syst. 2019, 165, 30–39

APPENDIX

1.1 OUTPUT SCREENSHOT

Screenshots of Model Training and Output




File Edit Selection View Go Run Terminal Help
app.py - Visual Studio Code
app.py
C: > Users > Krishna > Desktop > WebCam-Face-Emotion-Detection-main > app.py
from tensorflow.keras.utils import img_to_array
#from keras.preprocessing.image import img_to_array
from streamlit_webrtc import webrtc_streamer, VideoTransformerBase, RTCConfiguration, VideoProcessorBase, WebRtcMode
# load model
emotion_dict = {0:'angry', 1 :'happy', 2: 'neutral', 3:'sad', 4: 'surprise'}
# load json and create model
json_file = open('emotion_model1.json', 'r')
loaded_model_json = json_file.read()
json_file.close()
classifier = model_from_json(loaded_model_json)
# load weights into new model
classifier.load_weights("emotion_model1.h5")
#load face
try:
face_cascade = cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
except Exception:
st.write("Error loading cascade classifiers")
RTC_CONFIGURATION = RTCConfiguration({"iceServers": [{"urls": ["stun:stun.l.google.com:19302"]}]})
class Faceemotion(VideoTransformerBase):
def transform(self, frame):
img = frame.to_ndarray(format="bgr24")
#image gray
img_gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
faces = face_cascade.detectMultiScale(
image=img_gray, scaleFactor=1.3, minNeighbors=5)
for (x, y, w, h) in faces:
cv2.rectangle(img=img, pt1=(x, y), pt2=(
x + w, y + h), color=(255, 0, 0), thickness=2)
roi_gray = img_gray[y:y + h, x:x + w]
roi_gray = cv2.resize(roi_gray, (48, 48), interpolation=cv2.INTER_AREA)
if np.sum([roi_gray]) != 0:
Discovering Python Interpreters - Quokka
Ln 1, Col 1 Spaces: 4 UTF-8 CRLF Python 3.8.8 ('base': conda)
27°C
Partly cloudy
Search
ENG
IN
18:08
20-04-2023